

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

FILING OF A UNITED STATES PATENT APPLICATION

SYSTEM FOR HANDLING MULTIPLE DISCRETE COSINE TRANSFORM MODES
AND METHOD THEREOF

INVENTORS:

Daniel W. Wong
7601 Bathurst Street
Thornhill, Ontario L4J 4H5
Canada

Milivoje Aleksic
4 Harmony Hill Crescent
Richmond Hill, Ontario L4C 8Z1
Canada

Wayne Y.J. Wu
752 Peter Robertson Blvd.
Brampton, Ontario, L6R 1V2
Canada

William Hui
243 Highglen Avenue
Markham, Ontario L3S 3W3
Canada

ATTORNEY OF RECORD
J. GUSTAV LARSON

SIMON, GALASSO & FRANTZ, PLC
P.O. Box 26503
Austin, TX 78755-0503
PHONE (512) 336-8957
FAX (512) 336-9155

Express Mail Label No. EL855711007US

Date of Deposit: 1-17-02

I hereby certify that this paper is being deposited with the U.S. Postal Service
"Express Mail Post Office to Addresses" service under 37 C.F.R. Section 1.10 on
the 'Date of Deposit', indicated above, and is addressed to the Commissioner of
Patents and Trademarks, Washington, D.C. 20231.

Name of Depositor: Terri Alloway

(print or type)

Signature: Terri Alloway

SYSTEM FOR HANDLING MULTIPLE DISCRETE COSINE TRANSFORM MODES AND METHOD THEREOF

FIELD OF THE DISCLOSURE

The present invention relates generally to processing video and more particularly to
5 processing motion compensation error data.

BACKGROUND

Digital video is generally processed in sets of video frames. Each frame is a still image representing an instant in time of the video being processed. These frames include a large amount of data that must be transmitted to a display device. In order to facilitate transmission, each frame is further broken down into blocks relating to 8x8 picture elements. The blocks are individually transmitted and then recombined to form a frame for display. The amount of data needed to represent these image blocks may still be quite large however, so motion compensation is sometimes used to reduce the amount of data needed to represent the image blocks.

Using motion compensation, image blocks can be represented by motion compensation
15 vectors and error data. Motion compensation vectors are used in prediction frames, also known as
“P-frames”. P-frames allow an object in one frame to simply be repositioned in a new frame.
Accordingly, the image blocks used to represent the object in the new frame may be processed with
motion vectors, using the image blocks in the original frame as reference. The motion vectors
provide the direction and distance in which the referenced image blocks have moved in the new, or
20 predicted, frame.

In some cases, motion compensation vectors are all that are needed to reproduce an image block; however, in many situations, some other differences exist between the reference image block

and the block in the predicted frame. Error data can be used to recover the differences, and adequately generate the image block. The error data itself is basic image information, including the luminance of the pixels within the image block. A transform, such as a discrete cosine transform (DCT), can reduce the amount of error data in a transformed data set. This transformed data set 5 includes transfer coefficients which can be inverse transformed to reproduce the error data.

In some cases, no motion vectors can be generated for a given image block. For example, when a video switches to a completely new scene, none of the objects in the new frame can be referenced to objects in the previous frame. In such a case, the image block is represented only with error data. Furthermore, some reference frames for motion compensation are made up of image 10 blocks represented with only error data. These reference frames including only error data are referred to as intra-frames, or I-frames. The P-frames are motion compensated frames that use previous I- or P-frames for reference.

In addition to P-frames and I-frames, bi-directional frames (B-frames) may be used in handling image data. Bi-directional frames use previous or upcoming I- or P-frames for reference. It should be noted that B-frames are never used as reference themselves to avoid the accumulation of precision errors.

Digital video decoding hardware is used to process the error data and motion compensation vectors into video frame data. To generate the video frame data the motion compensation vector data and the error data are captured. The transformed error data sets are inverse transformed, such as 20 through an inverse discrete cosine transform (IDCT) component, to reproduce the error data. In conventional systems, 8x8 blocks of transformed error data are sent to be inverse transformed one at a time. The conventional video decoding hardware uses an 8x8 IDCT component to reproduce a block of error data.

To facilitate error data processing of image data containing interlaced video, separate fields 25 may be sent separately for processing. Two 4x8 sets of transformed error data are sent to hardware for processing into an 8x8 block of error data. Each 4x8 set is generally unique to a field of a video

frame. To inverse transform the two 4×8 (2-4-8) sets of transformed error data into an 8×8 block of error data, conventional systems use separate hardware components for processing the 8×8 transformed error data.

Conventional video encoding/decoding systems must implement separate components for handling processing for 8-8 versus 2-4-8 data and for transforming and inverse transforming data. Implementing separate components is costly and reduces the amount of space available for implementing other hardware components. From the above discussion it is apparent that a method and system with improved efficiency for transforming and inverse transforming video data would be useful.

BRIEF DESCRIPTION OF THE DRAWINGS

Specific embodiments of the present invention are shown and described in the drawings presented herein. Various objects, advantages, features and characteristics of the present invention, as well as methods, operation and functions of related elements of structure, and the combination of parts and economies of manufacture, will become apparent upon consideration of the following description and claims with reference to the accompanying drawings, all of which form a part of this specification, and wherein:

FIG. 1 is a block diagram illustrating a system for processing video data, according to one embodiment of the present invention;

FIG. 2 is a block diagram illustrating components of a system for processing multiple forms of image data, according to one embodiment of the present invention;

FIG. 3 is a flow diagram illustrating a method of processing multiple forms of image data, according to one embodiment of the present invention; and

FIG. 4 is a block diagram illustrating a pipeline for calculating forward and inverse discrete cosine transforms on a set of image data using a single matrix structure, according to one embodiment of the present invention.

DETAILED DESCRIPTION OF THE FIGURES

Referring now to FIG. 1, a block diagram illustrating a system for processing video data is shown, according to one embodiment of the present invention. A video application 105 generates video data. A software driver 105 stores error data associated with the video data in error data buffer 112. The stored error data is passed to a discrete cosine transform (DCT) engine, such as DCT component 122. In one embodiment, the stored error data includes raw image data that may be used to generate a video frame. In one embodiment, transformed error data associated with a forward discrete cosine transform (FDCT) is processed by DCT component 122 using an inverse discrete cosine transform (IDCT) and stored in DCT results 134 of memory 130. Alternatively, non-transformed error data is processed by DCT component 122 using an FDCT. In one embodiment, DCT component 122 is further capable of processing error data associated with both an 8-8 block of image data and a 2-4-8 block of image data.

Video application 105 can include video applications such as digital video disk (DVD) player software, a digital television tuner, an application programming interface (API), a video capture application, or video encoding or decoding software. In one embodiment, when using motion compensation techniques to display video images, video information related to a new block of image data within a frame of video is temporal-wise compressed using motion compensation (MC) vectors. In temporal-wise compression, blocks in a new frame are compared to blocks in a reference frame. Objects in the reference frame may simply move or be displaced in the new frame. Therefore, an MC vector, indicating the direction and distance an object in a reference frame has moved, can be used to describe where the blocks representing the object should be in a new frame. MC vectors may not always be enough to represent the block in the new, or predicted, frame. Differences between the block in the reference frame and the block in the new frame are transmitted as error data.

5 Error data is generally image data, including pixel information to reproduce any image information not covered using MC vectors. The error data can be compressed using a DCT. The DCT is a discrete orthogonal transformation between a time and frequency domain. Generally an FDCT is performed on the error data to generate transformed error data coefficients, allowing an IDCT to later be used on the transformed error data coefficients to restore the error data from the DCT results. The error data can correct for any image information left out using the MC vectors. It should be noted that some blocks, even in predicted frames, may be sent using only transformed error data, without any corresponding MC vectors.

10 Error and MC data can be received through video application 105. Video application 105 can be an application programming interface (API), or a device driver interface (DDI), such as a DirectX Video Acceleration API/DDI. The video data, error data and MC vector data received by video application 105, are sent to a software driver, such as software driver 110. As previously discussed, in one embodiment, video application 105 includes video encoding software. Accordingly, video application 105 may generate non-transformed error data to be processed into transformed error data through DCT 122. Alternatively, video application 105 may include video decoding software in which video application 105 may provide transformed error data to be inverse transformed by an IDCT transform of DCT 122.

20 Software driver 110 receives the video data provided through video application 105. Error data is stored in an error data buffer 112. MC vector data is stored in MC buffer 114. A control 118 is used to monitor requests from graphics chip 120. When components 122 and 124 have completed processing a set of data, interrupts are sent to software driver 110, through control 118, indicating components 122 and 124 are ready to receive new data. In one embodiment, identifiers indicating a portion of a video to which a particular set of error data corresponds is sent to DCT component 122 with the set of error data, to allow processed error data to be matched with MC vector data for 25 processing.

Once error data with a particular identifier is received by DCT component 122, the identifier is stored in an identifier register 132. When the error data is processed through DCT component

122, an interrupt indicating the particular identifier is sent to control 118. Software driver 110 can use the reported identifier to send corresponding MC vector data from MC buffer 114. In one embodiment, software driver 110 sends all the sets of MC vector data in MC buffer 114 until it finds the set of MC vector data associated with the identifier. Alternatively, a semaphore may be used to track the error data processed from error data buffers 112 and MC vector data stored in MC buffer 115.

Graphics chip 120 includes components for processing video data from software driver 110. A DCT component 122 is used to process error data. In one embodiment, software driver 110 sends the error data. In another embodiment, the error data is read from memory 130 by DCT component 122. In one embodiment, DCT component 122 includes a DCT reader for receiving the error data, a DCT core for processing the error data and a DCT writer for storing the results from the DCT core. The DCT results may be stored in memory 130, such as in DCT results 134. DCT component 122 uses different DCT matrices for processing error data received as either 8-8 image data or as 2-4-8 image data. The DCT matrices are used to perform an FDCT on untransformed error data.

To process error data associated with transformed error data, the matrices are transposed, allowing an IDCT to be performed. The same DCT matrix is used in FDCT and IDCT operations; however, the DCT matrix is accessed differently to achieve a transposed DCT matrix for performing IDCT operations. In one embodiment, the transpose is performed by DCT component 122 by switching from a row-major access of the matrices to a column-major access of the matrices. For example, to perform an FDCT on 8-8 image data, DCT component 122 performs matrix multiplication on an 8x8 matrix formed by the 8-8 image data and a row-major accessed 8-8 DCT matrix. To perform an IDCT on 8-8 transformed image data, DCT component 122 performs matrix multiplication on an 8x8 matrix formed by the 8-8 transformed image data and a column-major accessed 8-8 DCT matrix. Similarly, a set of 2-4-8 matrices is used for processing 2-4-8 image data sets. DCT component 122 stores the results of the processed image data in DCT results 134, of memory 130.

In one embodiment, MC vector data sets stored in MC buffer 114, corresponding to the

processed DCT data stored in DCT results 134, are sent by software driver 110 to a motion compensation processing component, such as 3D pipe 124. In one embodiment, 3D pipe 124 receives a memory address with the MC vector data sets to indicate where to read the error data, stored in DCT results 134, related to the MC vector data sets. Alternatively, error data from DCT component 122 is sent to 3D pipe 124. 3D pipe 124 processes the MC vector data along with the corresponding error data to generate a complete set of image data. Sets of image data corresponding to inverse transformed data can be stored in frame buffer 136. Frame buffer 136 can be represented by a location in memory 130 or in hardware, such as in graphics chip 120. Alternatively, the set of image data can be delivered to a display device (not shown). Sets of image data corresponding to transformed data may be stored in memory 130 or output to an alternate set of video processing hardware (not shown). In one embodiment, a prediction plane is obtained based on the motion compensation vector data and a reference frame. The prediction plane may be combined with error data to produce the final image blocks.

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
988
989
989
990
991
992
993
994
995
996
997
998
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1

multiple forms of image data is shown, and is referenced generally as DCT component 122, according to one embodiment of the present invention. Image data 205 is read and processed through a transform engine, such as DCT component 122. Image data 205 is used to refer to raw image data that may result from error data processed through motion compensation. DCT component 122 includes a DCT reader 210 for receiving and preparing image data 205. Prepared image data is processed through DCT core 220. DCT core 220 applies a DCT or IDCT matrix to process the prepared image data. DCT core 220 is capable of performing both inverse and forward DCT processing. Accordingly, wither transformed or inverse transformed image data may be presented to input 211 for processing, depending on a particular mode of operation. Processed image data is passed to DCT writer 270, which stores the processed image data in memory (not shown). DCT reader 210, DCT core 220 and DCT writer 270 work together to allow DCT component 122 to read image data and store transformed, or inverse transformed, results in memory.

Input 211 of DCT reader 210 reads image data 205 from memory or an image data buffer, such as image data buffer 112 (FIG. 1). In one embodiment, DCT reader 210 supplies an address of memory to access image data 205. In another embodiment, DCT reader 210 sends a request for image data 205 from a buffer. For example, DCT reader 210 may generate an interrupt to software driver 110 (FIG. 1), requesting more image data from image data buffer 112 (FIG. 1). Input 211 may include pointers to the image data buffer or portion of memory from which image data 205 is to be read.

20 In one embodiment, image data 205, corresponding to transformed image data, may be encrypted. DCT reader 210 may use decryption component 212 to decrypt image data 205 prior to processing through DCT core 220. Image data 205 may be encrypted using a dynamic encryption key, wherein the dynamic encryption key value changes during transmission. Accordingly, DCT reader 210 may need to synchronize to a source device which is encrypting the data. In one embodiment, input 211 synchronizes to software driver 110 for decrypting image data 205. As previously discussed, identifiers may also be sent with image data 205, indicating a portion of an image frame to which they correspond. An interrupt is generated indicating, to a software driver, the corresponding MC vector data which needs to be sent, allowing related image data and MC vector

data to be processed together in a separate component, such as 3D pipe 124 (FIG. 1). Input 211 maintains synchronization of the current portion of an image frame being processed through the identifiers.

In one embodiment, image data corresponding to U- and V-plane image data is sent together.

5 Image data is broken down into Y, U, and V data. The Y data refers to a luminance, or intensity, associated with a source image. The U and V data represent chrominance, or color, planes associated with the source image. Since the human eye is more sensitive to changes of intensity than color, more luminance data is sent than U and V data for every image macroblock of a video frame being encoded. The U- and V-plane data is generally sent together as UV image data. To properly
10 process the chrominance data, the U and V data must be separated prior to processing by DCT core 220. Input 211 stores the V-plane image data in a V-plane buffer 214. The U-plane data is then sent to DCT core 220. Once the U-plane data is processed, the V-plane data from V-plane buffer 214 is sent to DCT core 220.

In one embodiment, image data 205, corresponding to transformed image data, is run-level encoded. A block of image data may be composed of several zeros with few non-zero values. To compress the image data, run-level coding sends the non-zero values with information regarding the number of zeros between subsequent non-zero values. For image data 205 to be processed correctly by DCT core 220, input 211 decodes run-level coded image data into a proper image data block for processing through DCT processor 225 of DCT core 220. It should be appreciated that other encoding and decoding algorithms may be used for compressing and decompressing the image data. For example, Huffman coding may be used to break down the image data into code words received and decoded by input 211. Alternatively, the image data may be compressed through block truncation coding. Other techniques of encoding and decoding the image data may be used without departing from the scope of the present invention.

25 In one embodiment, end-of-block (EOB) instructions are embedded in image data 205, where the image data corresponds to transformed image data. The EOB instructions are used by input 211 to indicate when a block of data ends. Input 211 responds to the EOB instruction by applying zeros

to the remaining portion of the received data block not received. In one embodiment, if a faulty transmission/reception of image data 205 causes an EOB instruction to not be received by input 211, DCT component 122 may become hung-up waiting for more image data. Accordingly, input 211 may complete incomplete blocks of image data when no EOB instruction is sent for a specified 5 period of time.

As previously discussed, the image data may relate to non-transformed image data, which is to be processed with a DCT, such as through DCT processor 225. The image data may also relate to transformed error data, which is to be processed through an IDCT, through DCT processor 225. Input 211 notifies table access component 226 of DCT processor 225 with the type of processing 10 (DCT or IDCT) to be performed. In one embodiment, input 211 identifies the type of processing to be performed through an indicator sent with image data 205. Image data processed through input 211 is stored in a buffer of DCT buffers 260, such as DCT coefficients buffers 265 or 266. Each of DCT coefficients buffers 265 and 266 hold a full block of image data for DCT processor 225.

DCT core 220 performs the DCT or IDCT processing on the image data stored in DCT buffers 269, such as first DCT coefficients buffer 265, received through input 211. As previously discussed, the image data may be 8-8 image data or 2-4-8 image data. 8-8 image data includes a single 8x8 block of image data to be processed. In comparison, 2-4-8 image data includes two related 4x8 sets of image data corresponding to separate fields of an interlaced image block.

In one embodiment, the image data is 8-8 image data. The 8-8 image data is processed using 20 an 8-8 DCT matrix 230. The 8-8 DCT matrix 230 includes values so that, when a block of 8-8 image data is multiplied by 8-8 DCT matrix 230, a one-dimensional DCT transformed result may be generated, as described further in reference to FIG. 4. In one embodiment, the one-dimensional result represents a set of first pass results. The same process is then repeated with a transpose performed on the first pass results to complete a two-dimensional DCT operation. In one 25 embodiment, when the DCT processor 225 is instructed that an IDCT is to be performed on 8-8 image data passed from DCT reader 210, the 8-8 image data is multiplied by a transpose of 8-8 DCT matrix 230. In one embodiment, to transpose 8-8 DCT matrix 230, 8-8 DCT matrix 230 is accessed

by table access component 226 using a column major scheme (column versus row) to perform an IDCT, while 8-8 matrix 230 is accessed by table access component 226 using a row major scheme (row versus column) for DCT operations.

5 In one embodiment, the image data is 2-4-8 image data. A 2-4-8 DCT matrix 240 is first applied to the 2-4-8 image data to generate a set of first pass results. The 8-8 DCT matrix 240 may then be applied to the first pass results to complete a two-dimensional IDCT or DCT operation. In one embodiment, DCT processor 225 is set, through input 211, to perform an FDCT on the 2-4-8 image data. Accordingly, a block of 2-4-8 image data is stored together as a matrix in a buffer of buffers 260, such as second DCT coefficients buffer 266. As previously discussed, 2-4-8 image data is constructed to include two 4x8 sets of data including information about fields of a block of video. A first 4x8 block includes a summation of the two fields. A second 4x8 block includes a difference between the two fields. The two 4x8 fields may be stored together as a full block of data. The matrix of the 2-4-8 image data stored in second DCT coefficients buffer 266 is multiplied by 2-4-8 DCT matrix 240, accessed through table access component 226, to generate transformed image data. 10 In one embodiment, the 2-4-8 DCT matrix 240 is accessed row-major to perform an FDCT operation and is accessed column-major for IDCT operations. Instead of transposing the DCT matrices 230 and 240, values from DCT matrices 230 and 240 may be read in a vertical direction when processing with the DCT matrices 230 or 240, with results being written in a horizontal direction. In one embodiment, 2-4-8 image data is converted into 8-8 image data and then processed as 8-8 image data.

20 DCT core 220 includes DCT buffers 260 for storing image data before and after being processed. First and second DCT coefficients buffers 265 and 266 are used for storing blocks of image data for processing through DCT processor 225. First and second DCT result buffers 261 and 262 may be used for storing the results of processing performed on image data stored in first and second DCT coefficients buffers 265 and 266, respectively. A third DCT result buffer 263 may be used to provide additional storage during processing. In one embodiment, third DCT result buffer 263 is used to merge results from two processed blocks, such as from first and second DCT results buffers 261 and 262, for combining processed U-plane and V-plane data to generate a single UV-

plane data set.

The DCT-processed (FDCT or IDCT) results from DCT results 261, 262 or 263 may be sent to DCT writer 270 for storage in memory. DCT writer 270 provides memory control to store the DCT-processed results in memory. DCT writer 270 makes appropriate memory requests from a memory controller (not shown) for storing the results. In one embodiment, buffers 260 are used to combine a first and second set of DCT-processed Y data, and a third and fourth set of DCT-processed Y-data prior to being sent to memory, allowing DCT writer 270 to utilize more memory bandwidth than sending each set of Y-data separately. In one embodiment, DCT component 122 is part of a single monolithic semiconductor device.

卷之三

Referring now to FIG. 3, a flow diagram illustrating a method of handling the processing of multiple forms of image data is shown, according to one embodiment of the present invention. Image data is read and processed through a DCT matrix. A different DCT matrix is used for processing different types of image data. In one embodiment, the DCT matrix is transposed for IDCT operations.

In step 310, a DCT processing component, such as DCT component 122 (FIG. 2), reads a set of image data. In one embodiment, the image data set is read from memory. Memory requests are sent to a memory controller, including a specific memory address to access the image data from. In another embodiment, the image data set is read from an image data buffer. The image data may be located in hardware, such as graphics chip 120 (FIG. 1), or be provided by a software driver, such as 20 from image data buffer 112 of software driver 110 (FIG. 1).

In step 315, it is determined if the data is protected. In one embodiment, image data corresponding to transformed image data is protected through encryption or another encoding process. If the image data is being read from the source, such as software driver 110, an encrypted link may be established using a dynamic encryption key. In step 315, if the image data is protected,

the image data is decoded in step 317. If the image data is not encrypted, or once protected image data has been decoded through step 317, the DCT processing component continues at step 320.

In step 320, it is determined if the image data is related to 8-8 image data or 2-4-8 image data. As previously discussed, 8-8 image data refers to an 8x8 block of image data relating to a full block of image data as a whole set. A 2-4-8 block of image data includes two separate 4x8 blocks of image data which are each generated using separate fields of a block of image data. A first 4x8 data set is associated with a summation of the two fields. A second 4x8 data set is associated with a difference between the two fields. In one embodiment, identifiers are attached with the image data to indicate whether the image data relates to 8-8 image data or 2-4-8 image data.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95

In step 320, if the image data is 8-8 image data, an 8-8 DCT matrix is selected for both passes in the two pass processing used to perform two-dimensional DCT calculations, as in step 330. Alternatively, if the image data is 2-4-8 image data, a 2-4-8 DCT matrix is selected for a first pass, as in step 250. In the case of 2-4-8 image data, the 8-8 DCT matrix would be selected to perform processing for a second pass. In one embodiment, the identifier included with the image data indicates whether the image data is related to 8-8 image data or 2-4-8 image data. in one embodiment, 2-4-8 image data is converted into 8-8 image data and then processed as 8-8 image data.

In step 340, it is determined if the image data is to be forward processed (FDCT) or inverse processed (IDCT). In step 340, if an FDCT is to be performed, the selected DCT matrix (from step 20 320) is left “as is” and the DCT processing component continues to step 370. In step 345, if an IDCT is to be performed, the selected DCT matrix is transposed, as in step 345. In one embodiment, the selected DCT matrix is read in a normal, row-major fashion for FDCT processing. To transpose the matrix for IDCT processing, the selected DCT matrix is read in a column-major fashion. Accordingly, step 345 may be used to simply apply an option to read, or index, the selected DCT 25 matrix by columns versus rows. In one embodiment, values from the selected DCT matrix are read in a vertical direction with results being written in a horizontal direction.

In step 370, the image data is processed using the selected DCT matrix. The selected DCT matrix (transposed if an IDCT operation) is multiplied by a matrix composed of the image data. In step 380, the DCT processing component outputs the processed image data. In one embodiment, the processed image data is output to memory.

5 Referring now to FIG. 4, a block diagram illustrating a pipeline for calculating forward and
inverse transforms on a set of image data using a single matrix structure is shown, according to one
embodiment of the present invention. Image data is processed using DCT transforms (FDCT or
IDCT). Received image data is related to either 8-8 image data or 2-4-8 image data. Either type of
image data is used to construct an 8x8 matrix of image data. A DCT matrix is constructed so that a
10 product of the DCT matrix and the image data matrix, results in an FDCT of the image data. To
generate an IDCT of the image data, a transpose of the DCT matrix is used.

An FDCT may be constructed using an FDCT function, $F(u,v)$. The two-dimensional FDCT represents a matrix of frequencies based on the input image data. In the function, $F(u,v)$, “ u ” denotes frequency values of the image data in a horizontal direction and “ v ” represents frequency values of the image data in a vertical direction. A function $f(x,y)$ may be used to represent the image data. In the function $f(x,y)$, “ x ” represents a horizontal position and “ y ” represents a vertical position of a particular picture element. Alternatively, the function $f(x,y)$ may represent an IDCT function performed on a transformed set of image data, $F(u,v)$. The functions of $F(u,v)$ and $f(x,y)$ for 8-8 image data, $F_{8,8}(u,v)$ and $f_{8,8}(x,y)$, are shown in the following equations:

$$F_{8-8}(h, v) = C(v)C(h)\sum_{y=0}^7 \sum_{x=0}^7 \cos\left(\frac{\pi v(2y+1)}{16}\right) \cos\left(\frac{\pi h(2x+1)}{16}\right) f(x, y) \quad \text{where} \quad \begin{aligned} C(h) &= 0.5/\sqrt{2} & \text{for } h = 0 \\ C(h) &= 0.5 & \text{for } h > 0 \\ C(v) &= 0.5/\sqrt{2} & \text{for } v = 0 \\ C(v) &= 0.5 & \text{for } v > 0 \end{aligned}$$

$$f_{8-8}(x, y) = \sum_{v=0}^7 \sum_{h=0}^7 C(v)C(h) \cos\left(\frac{\pi v(2y+1)}{16}\right) \cos\left(\frac{\pi h(2x+1)}{16}\right) F(h, v)$$

In one embodiment, weighting values are applied to each of the values of the result of $F(u,v)$. For 8-8 image data, the results of $F_{8,8}(u,v)$ are multiplied by the weighted values of $w_{8,8}(u,v)$ using values as

follows:

$$\begin{aligned}
 w_{8-8}(0,0) &= \frac{1}{4} & w(0) &= 1 & w(4) &= \frac{7}{8} \\
 w_{8-8}(h, v) &= \frac{w(h)w(v)}{2} \quad \text{where} & w(1) &= \frac{\cos(4\pi/16)}{4\cos(2\pi/16)\cos(7\pi/16)} & w(5) &= \frac{\cos(4\pi/16)}{\cos(3\pi/16)} \\
 & & w(2) &= \frac{\cos(4\pi/16)}{2\cos(6\pi/16)} & w(6) &= \frac{\cos(4\pi/16)}{\cos(2\pi/16)} \\
 & & w(3) &= \frac{1}{2\cos(5\pi/16)} & w(7) &= \frac{\cos(4\pi/16)}{\cos(\pi/16)}
 \end{aligned}$$

Similarly, the functions of $F(u,v)$ and $f(x,y)$ for 2-4-8 image data, $F_{2-4-8}(u,v)$ and $f_{2-4-8}(x,y)$, are shown in the following equations, where “z” denotes $y/2$, for indexing image data between alternating fields:

$$\begin{aligned}
 F_{2-4-8}(h, v) &= C(v)C(h)\sum_{z=0}^3 \sum_{x=0}^7 \cos\left(\frac{\pi v(2z+1)}{8}\right) \cos\left(\frac{\pi h(2x+1)}{16}\right) (f(x, 2z) \oplus f(x, 2z+1)) \\
 f_{2-4-8}(x, y) &= \sum_{v=0}^3 \sum_{h=0}^7 C(v)C(h) \cos\left(\frac{\pi v(2z+1)}{8}\right) \cos\left(\frac{\pi h(2x+1)}{16}\right) (F(h, v) \otimes F(h, v+4))
 \end{aligned}
 \quad \text{where} \quad
 \begin{aligned}
 \oplus & \text{ is } + \quad \text{for } v < 4 \\
 \oplus & \text{ is } - \quad \text{for } v > 3 \\
 \otimes & \text{ is } - \quad \text{for odd } y \\
 \otimes & \text{ is } + \quad \text{for even } y
 \end{aligned}$$

If the DCT coefficients $F_{2-4-8}(u,v)$ are weighted, the DCT coefficients are weighted differently (compared with standard DCT):

$$\begin{aligned}
 w_{2-4-8}(0,0) &= \frac{1}{4} \\
 w_{2-4-8}(h, v) &= \frac{w(h)w(2v)}{2} \quad \text{for } v < 4 \\
 w_{2-4-8}(h, v) &= \frac{w(h)w(2(v-4))}{2} \quad \text{for } v \geq 4
 \end{aligned}$$

In one embodiment, the functions of the FDCT are compressed by quantizing the values. A table of quantization values is used to limit the range and size of the results. For an IDCT, the values must be de-quantized prior to applying the IDCT. In one embodiment, the DCT functions, $F(u,v)$ for FDCT and $f(x,y)$ for IDCT, are applied using a DCT matrix. In one embodiment, a one-dimensional

8-8 DCT matrix for FDCT operations on 8-8 image data is as shown in Table 1.

	U=0	U=1	U=2	U=3	U=4	U=5	U=6	U=7
X=0	0.35355339	0.49039264	0.46193977	0.41573481	0.35355339	0.27778512	0.19134172	0.09754516
X=1	0.35355339	0.41573481	0.19134172	-0.09754516	-0.35355339	-0.49039264	-0.46193977	-0.27778512
X=2	0.35355339	0.27778512	-0.19134172	-0.49039264	-0.35355339	0.09754516	0.46193977	0.41573481
X=3	0.35355339	0.09754516	-0.46193977	-0.27778512	0.35355339	0.41573481	-0.19134172	-0.49039264
X=4	0.35355339	-0.09754516	-0.46193977	0.27778512	0.35355339	-0.41573481	-0.19134172	0.49039264
X=5	0.35355339	-0.27778512	-0.19134172	0.49039264	-0.35355339	-0.09754516	0.46193977	-0.41573481
X=6	0.35355339	-0.41573481	0.19134172	0.09754516	-0.35355339	0.49039264	-0.46193977	0.27778512
X=7	0.35355339	-0.49039264	0.46193977	-0.41573481	0.35355339	-0.27778512	0.19134172	-0.09754516

Table 1. DCT matrix for 8-8 image data

The DCT matrix is multiplied by the 8-8 image data to generate one-dimensional FDCT results. It should be noted the DCT matrix of Table 1 may also be used to generate IDCT results through a transpose of the DCT matrix of Table 1. The transposed DCT matrix is then multiplied by the image data. Similarly, a DCT matrix may be constructed for processing 2-4-8 image data. An example of a DCT matrix for 2-4-8 data is shown in Table 2.

	U=0	U=1	U=2	U=3	U=4	U=5	U=6	U=7
X=0	0.35355339	0.46193977	0.35355339	0.19134172	0.35355339	0.46193977	0.35355339	0.19134172
X=1	0.35355339	0.46193977	0.35355339	0.19134172	-0.35355339	-0.46193977	-0.35355339	-0.19134172
X=2	0.35355339	0.19134172	-0.35355339	-0.46193977	0.35355339	0.19134172	-0.35355339	-0.46193977
X=3	0.35355339	0.19134172	-0.35355339	-0.46193977	-0.35355339	-0.19134172	0.35355339	0.46193977
X=4	0.35355339	-0.19134172	-0.35355339	0.46193977	0.35355339	-0.19134172	-0.35355339	0.46193977
X=5	0.35355339	-0.19134172	-0.35355339	0.46193977	-0.35355339	0.19134172	0.35355339	-0.46193977
X=6	0.35355339	-0.46193977	0.35355339	-0.19134172	0.35355339	-0.46193977	0.35355339	-0.19134172
X=7	0.35355339	-0.46193977	0.35355339	-0.19134172	-0.35355339	0.46193977	-0.35355339	0.19134172

Table 2. DCT matrix for 2-4-8 image data

The DCT matrix of Table 2, for 2-4-8 image data, is applied on an 8x8 block consisting of 2-4-8 image data through matrix multiplication, as discussed for 8-8 image data and the DCT matrix of Table 1. Similarly, the DCT matrix of Table 2 is transposed to perform IDCT operations on the 2-4-8 image data. As can be seen in Tables 1 and 2, common sets of coefficients are used in both DCT matrices. As shown in Table 3, each of the coefficient values may be approximated through a summation of three integer-based fractions.

	1st	2nd	3rd	Error
0.49039264	7/16	7/128	-7/4096	0.000085875
0.46193977	7/16	3/128	1/1024	-0.000025704
0.41573481	7/16	-3/128	7/4096	0.000036678
0.35355339	1/4	7/64	-3/512	-0.000037766
0.27778512	1/4	7/256	7/16384	-0.000014120
0.19134172	3/16	1/256	-1/16384	0.000003499
0.09754516	3/32	1/256	-1/8192	-0.000010981

Table 3. DCT Coefficient Approximations

Using the approximations of Table 3, the DCT matrices of Tables 1 and 2 may be applied to respective 8-8 and 2-4-8 image data blocks through a four-stage pipeline, as shown in FIG. 4. A pipeline input 410 is used to input the image data values into the matrices. A 1X multiplier 420 is used for the numerator in the approximations shown in Table 3, such as the $\frac{1}{4}$ approximation listed in the 1st column. The 3X multiplier 430 and 7X multiplier 440 are also used for fractions involving a numerator of '3' and '7', respectively. The multipliers 420, 430 and 440 are used to generate the coefficients 450.

Multiplexers 460 may be used to select individual values from coefficients 450. Each value of an image data matrix is input through pipeline input 410 and multiplied by a selected coefficient of coefficients 450. The coefficient is selected using a POSITION signal 462, indicating a current position in the image data matrix. A counter (not shown) may be used to clock in image data values into pipeline input 410 and update the current value of POSITION signal 462. A DCT/IDCT signal 464 also is used to select between row-major and column-major interpretations of the DCT matrices.

An 8-8/2-4-8 signal 463 is used to select a type of DCT matrix needed for the current operations, such as selecting among an 8-8 matrix or a 2-4-8 DCT matrix. In one embodiment, accumulators 470 are used to combine the products of the coefficients 450 and the image data values. In one embodiment, an image data value is clocked into pipeline input 410 for every clock pulse; however, the corresponding results of the DCT operation (FDCT or IDCT) are only finalized in accumulators 470 after every eighth clock pulses. It should be noted that accumulators 470 combine previous product values as are combined in general matrix product operations. In one embodiment, the system of FIG. 4 is processed through software. Alternatively, the system of FIG. 4 may be processed through hardware.

The systems described herein may be part of an information handling system. The term "information handling system" refers to any system that is capable of processing information or transferring information from one source to another. An information handling system may be a single device, such as a computer, a personal digital assistant (PDA), a hand held computing device, 5 a cable set-top box, an Internet capable device, such as a cellular phone, and the like. Alternatively, an information handling system may refer to a collection of such devices. It should be appreciated that while components of the system have been described in reference to video and audio processing components, the present invention may be practiced using other types of system components. It should be appreciated that the system described herein has the advantage of providing FDCT and 10 IDCT operations for both 8-8 and 2-4-8 image data sets.

In the preceding detailed description of the embodiments, reference has been made to the accompanying drawings which form a part thereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical and electrical changes may be made without departing from the spirit or scope of the invention. To avoid detail not necessary to enable those skilled in the art to practice the invention, the description may omit certain information known to those skilled in the art. Furthermore, many other varied embodiments that incorporate the teachings of the invention may be easily constructed by those skilled in the art. Accordingly, the present invention is not intended to be limited to the specific form set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents, as can be reasonably included within the spirit and scope of the invention. The preceding detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.